

Module Graphite-Perfdata

Sommaire

[Explication](#)
[Configuration](#)

Explication

Le module Graphite-Perfdata permet d'envoyer et stocker les métriques dans un serveur Graphite (via Carbon). Il est possible de modifier des paramètres (comme l'adresse du serveur Graphite ou son port) via le fichier de configuration ci-dessous.

Configuration

Voici le fichier CFG de configuration présent dans : **/etc/shinken/modules/graphite.cfg**

```

=====
# Graphite-Perfdata
=====
# Daemons that can load this module:
# - broker
# This module send metrics into a graphite (carbon) server
=====

define module {

# Shinken Enterprise. Lines added by import core. Do not remove it, it's used by Shinken Enterprise to
update your objects if you re-import them.
    _SE_UUID          core-module-7d9e22e45adc11e5b3b3080027f08538
    _SE_UUID_HASH     2136b2756b5160cfad074e796ebc373e
# End of Shinken Enterprise part

    ===== Module identity =====
    # Module name. Must be unique
    module_name        Graphite-Perfdata

    # Module type (to load module code). Do not edit.
    module_type        graphite_perfdata

    ===== Workers in the broker =====
    # This module will use workers in the broker, each worker will manage a shard of all hosts/checks.
    # This parameter is used by the broker to set the number of workers. Each worker will use one CPU, which
will balance the metrology processing load among CPUs.
    # default: 1
    broker_module_nb_workers    1

    ===== Graphite address =====
    # host: graphite server address (ip or fqdn)
    host                        localhost

    # port: TCP port of the graphite server used for writing data
    port                        2003

    # webapp_port: TCP port of the graphite server used for reading data
    webapp_port                 80

    ===== realm filtering =====
    # By default, this module will save metrics from all realm and subrealms of the broker realm.
    # You can use realm_store_only to save only the realm you want into the graphite server
    #realm_store_only           Realm1, Realm2, Realm3

    ===== Storage tuning =====
    # For performance purpose, we can globally disable storage of warning and/or error thresholds
    # default: 1 => store_warnig_threshold
    # broker_module_graphite_perfdata_store_warning_threshold 1
    #
    # default: 1 => store error_threshold
    # broker_module_graphite_perfdata_store_error_threshold    1
}

```